

PATENT APPLICATION

Invention Title:

CONFIGURING NETWORK SETTINGS OF THIN CLIENT DEVICES USING PORTABLE
STORAGE MEDIA

Inventors:

Scott Manchester	US	Redmond	Washington
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY

Benjamin Nick	US	Seattle	Washington
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY

Trevor W. Freeman	UK	Sammamish	Washington
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY

Dalen Abraham	US	Duvall	Washington
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY

INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
-----------------	-------------	-------------------	--------------------------

INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
-----------------	-------------	-------------------	--------------------------

Be it known that the inventors listed above have invented a certain new and useful invention
with the title shown above of which the following is a specification.

CONFIGURING NETWORK SETTINGS OF THIN CLIENT DEVICES USING PORTABLE STORAGE MEDIA

FIELD OF THE INVENTION

[0001] This invention pertains generally to the field of computer networks and more particularly to a mechanism for simplifying the process of configuring nodes in computer networks.

BACKGROUND OF THE INVENTION

[0002] The use of data communication networks continues to grow. In small as well as large corporate settings, wired local area networks (LANs) and wide area networks (WANs) have become an established feature of conducting business, and wireless networks are being increasingly employed. The use of network technology in the home, both wired and wireless, is a more recent phenomenon and has been slower to develop. In addition to facilitating Internet connectivity, home networking permits personal computing devices and various consumer electronic devices and appliances within the home to communicate with each other. Wireless technology, such as IEEE 802.11 wireless networks and networks of Bluetooth-enabled devices, is attractive in home as well as corporate environments for reasons of convenience, mobility and flexibility.

[0003] A principal impediment to the wider adoption of wireless networking technology in the home and other non-corporate environments has been the difficulty experienced by non-expert users in configuring network devices. Configuring network devices often requires detailed knowledge of the hardware, software, and protocols of the network, and can be too complicated for an average user. Moreover, many network devices nowadays are "thin-client" devices that typically run a reduced version of an operating system and do not have support for the user interface components (e.g., a monitor, keyboard, and/or mouse) and functions of conventional personal computers. The lack of convenient and intuitive I/O capabilities often makes it difficult to configure a thin client device for a home network. For example, it can be troublesome for many users to configure a wireless access point (WAP), which is typically a

thin client device. It can be particularly laborious and difficult to modify the network settings of a thin client device to configure it for a wireless network, when the thin client device does not have a ready connection with a regular computer. With the advancements of wireless networking technologies, various types of thin client devices, such as digital audio receivers, wireless printers, set top boxes, etc., are added to home networks, and there is a need for a simple and easy way to configure such thin client devices for network connectivity.

BRIEF SUMMARY OF THE INVENTION

[0004] In accordance with the invention, the task of configuring a thin client device for networking functionality and connectivity is made particularly simple and easy by using a portable computer-readable media device, such as a USB flash drive or an SD memory card, to transfer and automatically load network settings into the thin client device. A configuration application on a personal computer aids the user in generating configuration data that include proper network settings for a wide-area network (WAN), local-area network (LAN), or wireless network, depending on the intended purposes and network functions of the thin-client device. The configuration data may also include device configuration information, security information, and file sharing information. The configuration application then generates an Extensible Markup Language (XML) file embodying the configuration data, and writes that XML file into the portable media device. A user may then install the portable media device in a thin client computing devices to transfer the configuration data to the device. The thin client device detects the connection of the portal media device and automatically loads the configuration data, and provides signals to indicate completion of the configuration operation. As a result, a thin client device may be provisioned with the network settings and other configuration data quickly and conveniently without the need for a full-fledged user interface for interacting with the user.

[0005] Additional features and advantages of the invention will be apparent from the following detailed description of illustrative embodiments which proceeds with reference to the accompanying figures.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] While the appended claims set forth the features of the present invention with particularity, the invention and its advantages are best understood from the following detailed description taken in conjunction with the accompanying drawings, of which:

[0007] Figure 1 is a schematic diagram illustrating an exemplary architecture of a computing device that may be used for carrying out a process of configuring a thin client computing device in accordance with an embodiment of the invention;

[0008] FIG. 2 is a schematic diagram illustrating an exemplary home network that has network connectivity for wide-area, local-area, and wireless networks and includes a plurality of thin client devices that may be configured in accordance with an embodiment of the invention;

[0009] FIG. 3 is a schematic diagram illustrating the use of a portable media device for transferring configuration data including network settings to a thin client computing device;

[0010] FIG. 4A-F are schematic diagrams illustrating user interface screenshots of a configuration application for generating configuration data for a thin client computing device and storing the configuration data on a portable media device for transfer to the thin client device;

[0011] FIG. 5 is a schematic diagrams illustrating a software architecture for generating network setting data for configuring network devices in accordance with an embodiment of the invention;

[0012] FIG. 6 is a flow diagram illustrating a method for configuring a thin client computing device using a portable media device in accordance with an embodiment of the invention;

[0013] FIG. 7 is a process diagram illustrating a method for configuring a thin client device for joining an ad hoc wireless network in accordance with an embodiment of the invention;

[0014] FIG. 8 is a process diagram illustrating a method for configuring a wireless access point in accordance with an embodiment of the invention;

[0015] FIG. 9 is a data structure diagram illustrating fields of an XML schema for representing wireless network configuration settings in accordance with an embodiment of the invention;

[0016] FIGS. 10A-C are data structure diagrams illustrating fields of an XML schema for representing WAN configuration settings in accordance with an embodiment of the invention; and

[0017] FIG. 11 is a data structure diagram illustrating fields of an XML schema for representing LAN configuration settings in accordance with an embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0018] Methods and systems for configuring network devices with a portable media device will now be described with respect to certain embodiments. The skilled artisan will readily appreciate that the methods and systems described herein are merely exemplary and that variations can be made without departing from the spirit and scope of the invention.

[0019] The present invention will be more completely understood through the following detailed description, which should be read in conjunction with the attached drawings. In this description, like numbers refer to similar elements within various embodiments of the present invention. The invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as procedures, being executed by a personal computer. Generally, procedures include program modules, routines, functions, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, and microprocessor-based or programmable consumer electronics devices. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices. The term computer system may be used to refer to a system of computers such as may be found in a distributed computing environment.

[0021] Figure 1 illustrates an example of a suitable computing system environment 100 in which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as

to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100. Although at least one embodiment of the invention does include each component illustrated in the exemplary operating environment 100, another more typical embodiment of the invention excludes some or all non-essential components, for example, input/output devices other than those required for network communications.

[0022] With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of the computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures.

[0023] The computer 110 typically includes a variety of computer-readable media. Computer-readable media can be any available media that can be accessed by the computer 110 and includes both volatile and nonvolatile media, and removable and non-removable media. By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 110. Communication media typically embodies computer-readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation,

communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. Combinations of the any of the above are included within the scope of computer-readable media.

[0024] The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. By way of example, and not limitation, Figure 1 illustrates operating system 134, application programs 135, other program modules 136 and program data 137.

[0025] The computer 110 may also include other removable and non-removable, volatile and nonvolatile computer storage media. By way of example only, Figure 1 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CDROM. Other computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, DVDs, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

[0026] The computer system may include interfaces for additional types of removable non-volatile storage devices. For instance, the computer may have a USB port 153 that can accept a USB flash drive (UFD) 154, or a SD card slot 157 that can accept a Secure Digital (SD) memory card 158. A USB flash drive is a flash memory device that is fitted with a USB connector that can be inserted into a USB port on various computing devices. A SD memory card is a stamp-sized flash memory device. Both the USB flash drive and SD card offer high storage capacity in a small package and high data transfer rates. Other types of removable storage media may also be used for implementing the invention.

[0027] The drives and their associated computer storage media, discussed above and illustrated in Figure 1, provide storage of computer-readable instructions, data structures,

program modules and other data for the computer 110. In Figure 1, for example, hard disk drive 141 is illustrated as storing an operating system 144, application programs 145, other program modules 146 and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers herein to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 110 through input devices such as a tablet, or electronic digitizer, 164, a microphone 163, a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 by way of an interface, such as a video interface 190. The monitor 191 may also be integrated with a touch-screen panel or the like. Note that the monitor and/or touch screen panel can be physically coupled to a housing in which the computing device 110 is incorporated, such as in a tablet-type personal computer. In addition, computers such as the computing device 110 may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through an output peripheral interface 194 or the like.

[0028] The computer 110 preferably operates or is adaptable to operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a peer device or other network node, and typically includes some or all of the elements described above relative to the computer 110, although only a memory storage device 181 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a LAN 171 and a WAN 173, but may also include other networks. For example, in the present invention, the computer 110 may comprise the source machine from which data is being migrated, and the remote computer 180 may comprise the destination machine. Note however that source and destination machines need not be initially connected by a network or otherwise, but instead, data may be migrated by way of any media capable of being written by the source platform and

read by the destination platform or platforms. For example, one non-limiting instance of such a medium is a portable flash memory medium.

[0029] When used in a LAN environment, the computer 110 is connectable to the LAN 171 through a network interface or adapter 170. The computer 110 may also include a modem 172 or other means for establishing communications over the WAN 173. The modem 172, which may be internal or external, may be connected to the system bus 121 by way of the user input interface 160 or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, FIG. 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0030] Turning to FIG. 2, the present invention is directed to a simple and convenient way for a user to set up thin client computing devices in a home network or the like. The term "thin client" commonly refers to a computing device that is quipped with a microprocessor but runs a simplified or reduced operation system (e.g., Windows CE of Microsoft Corporation or an "embedded" version of an operating system) and does not have or support conventional user interface I/O devices such as a monitor and a keyboard. As a result, thin client devices typically have very limited input means, such as buttons with predefined functions that a user can use to enter commands. Some of them may not even have any user-operable input means. They also typically have very limited signaling means, such as LED's as indicator lights, or a liquid crystal display (LCD) screen for displaying simple signals and messages. Due to the limited user interface I/O capabilities, it can be a very complex and laborious task to configure a thin client device. Conventionally, the task of configuring a thin client device is made easier by connecting the thin client device to a personal computer, and running a proprietary configuration program on the personal computer that interacts with the thin client device to change settings on the device. This approach becomes impractical for many new thin client devices for use in a network environment that rely on wireless transmissions for network communications. The present invention allows such thin client devices to be configured easily with desired network settings and other operation parameters with minimal user intervention.

[0031] By way of example, FIG. 2 shows an exemplary network environment that has various thin client devices. A user's personal computer (PC) 200 is connected to an Ethernet-based local area network (LAN) 202. Other devices connected to the Ethernet include, for example, other PC's 210, a printer 212, a networkable television 214, a networkable telephone 216. The LAN may include a wireless access point 218 for forming an infrastructure wireless network. The LAN may further include a residential gateway device 222 that is connected to an external wide area network (WAN) through, for instance, a broad-band modem 220 for accessing the Internet 212. The residential gateway device 222 may also have wireless transmission capabilities to enable it to function as a wireless access point to communicate with wireless computing devices.

[0032] The network environment may also include wireless networks. In an infrastructure wireless network 203, wireless computing devices communicate with each other through the wireless access point 218. An ad hoc wireless network 234 may also be formed among computing devices that communicate wirelessly in a peer-to-peer fashion without going through an access point. The wireless devices may include notebook computers 230, a tablet computing device 232, and various other types of wireless devices such as a wireless television 238, a cellular phone 240, a wireless printer 250, a media center extender 260, a pocket PC 262, a wireless picture frame 268, wireless speakers 270, a wireless media player 272, etc. Other types of existing and new wireless devices can be added to the wireless networks. Wireless devices may communicate utilizing any suitable wireless communication protocol. Examples of suitable wireless communication protocols include wireless communication protocols in accordance with the Institute of Electrical and Electronics Engineers (IEEE) 802.1x series of standards, the Bluetooth® group of standards and the Ultra-Wideband (UWB) group of standards. As can be seen, many of the devices in the wired and wireless networks are thin client devices and can be difficult or cumbersome to configure for networking and other functions in the conventional way.

[0033] In accordance with a feature of the invention, the process of configuring a thin client device is made very simple and easy by using a portable media device to transfer configuration data and initiate automatic configuration of the device. Referring to FIG. 3, a configuration program 322 on a computer 312 assists a user in generating the configuration data for the thin

client device 314. Once the configuration data are generated, they are stored on a portable media (PM) device 326 connected to the computer 312. The portable media device 326 may be, for example, a USB flash drive 328 that may be inserted into a USB port 330, or a SD memory card 334 that may be inserted into a SD card slot 336. It will be appreciated that the invention is not limited to these two portable memory media, and other types of portable media may be used for implementing the invention.

[0034] The configuration data generated by the configuration program 322 may include network settings for the thin client device to communicate other computing devices on the network. Depending on its network location and functions, the thin client device 314 may require network settings for a wireless network (infrastructure or ad hoc), a LAN, or a WAN, and sometimes all of them. For example, the residential gateway 222 in FIG. 2 may require wireless network settings when it functions as an access point, LAN settings for communications over the Ethernet, and WAN settings for communicating with the WAN 206 through the broadband modem, which may require its own special settings. In addition to network settings, the configuration data may include other parameters pertaining to the operations of the device, such as device-specific configuration information, security information, and file sharing information.

[0035] After the configuration data are stored on the portable media device 326, the portable media device is disconnected from the first computer 312 and used to transfer the configuration data to one or more thin client devices. In a preferred embodiment, to set up a thin client device, all the user has to do is to connect the portable media device 326 to that device. The thin client device 314 detects the connection of the portable media device, and may run a configuration program 344 to load the configuration settings from the portable media device 326 and automatically configure the thin client device using the received configuration data.

[0036] To further simplify the task of setting up a thin client device, the configuration program on the computer 312 provides a user interface to help a user to go through the steps of the process of defining the network settings and other configuration data for the thin client device. For simplicity of illustration, the following description will describe an example in

which the portable media device is a USB flash drive, and the task is to configure a thin client device for joining an ad hoc wireless network.

[0037] Exemplary user interface screens are shown in FIGS. 4A-4F. For illustration purposes, the portable media device used in the example illustrated by these UI screens is a USB flash drive. In the user interface screen 460 illustrated in FIG. 4A, the user is informed that the configuration program will help the user create the settings for the wireless network, and then the settings will be saved on the USB flash drive and used to configure other computers or thin client devices that are to join the wireless network. In the second UI screen 462 shown in FIG. 4B, the user are given the options of setting up an infrastructure wireless network and setting up an ad hoc wireless network, and the user selects to set up an ad hoc network.

[0038] In the UI screen 464 shown in FIG. 4C, the user is presented with a field 476 for selecting a name for the wireless ad hoc network, and a field 478 for selecting a network key. In one implementation, to minimize the work the user has to do, the configuration program may generate the network name and security key for the user, and the user may reject them and enter her own network name and key if she so desires. To that end, the name and key fields are initially filled by the wireless configuration program with values comprising either arbitrary data or data that is chosen using identifiers stored on the computer. For example, if the computer operating system has been registered to "John Smith", then the wireless configuration program can generate "JohnSmithsNetwork" as the default network name. The wireless configuration program may call a function of the operating system to generate a security key for the wireless network. The user can accept the network name and key suggested by the configuration program by clicking "Next", or can edit the fields manually. Additionally, by clicking an "Advanced" button 480, the user is permitted to edit a variety of other network configuration settings, which otherwise are automatically generated by the wireless configuration program. An advantage of providing a key generated by the computer is that the key may be a full-sized key containing randomized characters, for example, alphanumeric characters. Such a key may provide enhanced security as compared to a user-entered key, which tends to be short and contain patterns that are easy for the user to memorize. Since the network settings, including the security key, are to be transferred to other computers by means

of the portable media device, the user does not have to try to create a key that she can remember and is more likely to use the key generated by the computer.

[0039] In the UI screen 466 shown in FIG. 4D, the wireless configuration program prompts the user to insert and identify a USB flash drive. When the user clicks “Next” on this screen to indicate that the USB flash drive has been inserted, the wireless configuration program stores the generated network settings for the wireless ad hoc network to the USB flash drive.

[0040] When the network settings have been completely stored on the USB flash drive, the user is presented with another UI screen 468 shown in FIG. 4E, prompting the user to remove the portable media device and plug it into all additional computing devices that are to join the ad hoc network. The UI screen 468 also presents the user with a “Print” button 482, allowing the user to print a hard copy of the network settings, which allows the user to manually configure those network devices that do not accept the portable media device or that cannot be automatically configured for the network.

[0041] When the user uses the USB flash drive to set up another computing device for the wireless network, the settings established on that device are written back onto the USB flash drive. After the user has used the USB flash drive to set up other computing devices for the ad hoc wireless network, she returns to the first computer 212 and inserts the USB flash drive in a USB port of the computer. The configuration program reads the setting data written by the other devices and presents a UI screen 470 as shown in FIG. 4F. The UI screen 470 identifies the devices that have been successfully set up for the wireless network. In addition, the configuration program presents the option of removing the settings from the USB flash drive. In the UI screen 470 of FIG. 4F, this option is presented as a check box 472. This prevents the network settings being inadvertently revealed to others when the USB flash drive is later used to transfer data to other computers. Since some users may have the tendency to use the same network name or key, this option of removing the network settings is provided to provide enhanced security. Additional enhanced security measures may be incorporated into configuration processes such as the processes corresponding to FIGS. 4A-4F. For example, utilizing conventional cryptographic techniques, some or all of the settings stored on the portable media device may be encrypted and then require suitable authentication information before being decrypted and/or allow decryption a particular number of times or within a

particular date range. Examples of suitable authentication information include a personal identification number (PIN), a password, a pass-phrase and suitable biometrics such as a thumbprint.

[0042] Turning now to FIG. 5, a software architecture used in an embodiment of the invention for generating network settings and other configuration data is now described. A configuration program 502 is executed on a computer and communicates with the computer through a wireless configuration application programming interface (API) 504 to generate network configuration settings. In the Windows operating system environment of Microsoft Corporation, for example, the WZCDLG.DLL library can be used for generating wireless settings.

[0043] In accordance with a feature of the embodiment, the network settings and other configuration data for the wireless ad hoc network are stored in the format of an Extensible Markup Language (XML) file. The use of an XML file presents a standard format that can be recognized by many different devices. The configuration program 502 outputs the XML files to the computer through a provisioning API 506. The configuration program 502 further outputs XML files for writing into an attached portable media device, such as a USB flash drive 508. To that end, a flash configuration device driver 510 reads configuration files and writes a device configuration file to the USB flash drive when it is provisioned.

[0044] The configuration program 502 stores several files on the USB flash drive 508 for use in the network configuration process. In the embodiment illustrated in FIG. 5, the files include the XML files representing the generated network configuration settings. In one implementation, the XML file containing the wireless network settings are give a special extension name such as "wfc" as shown in FIG. 5, to indicate that the file contains wireless configuration settings. Thus, when the USB flash drive is plugged into another computing device, the operating system of that device will recognize the files 512 as containing wireless network setup information and will invoke the wireless configuration program on that device to handle the files. The USB flash drive 508 may store several XML files representing the generated network configuration settings. A WSETTING.XML file 512 contains settings for a wireless network. An LSETTING.XML file 522 contains settings for a LAN. An NSETTING.XML file 528 contains settings for a WAN. These XML files are described in

greater detail below. By storing network configuration settings for LAN, WAN and wireless networks, a single USB flash drive 508 can be used to configure a variety of devices (such as personal computers, routers, printers, PDAs, and WAPs) to communicate over a variety of types of networks. The USB flash drive may also be used to store device-specific configuration data, which preferably are in an XML file. In FIG. 5, the XML file containing device-specific configuration data is the DEVICE SETTING.WFC file 528.

[0045] In addition, a network setup application 514 (called "Downlevel Flash Config Wizard" in FIG. 5) may be stored on the USB flash drive 508 in order to facilitate the configuring of network settings for other devices. When the USB flash drive 508 is attached to another device, that device can run the network setup application to load the relevant network settings from the USB flash drive 508 to the other device.

[0046] As also shown in FIG. 5, the USB flash drive 508 may be used to store device configuration log files 516 that are written into the flash drive by devices that provisioned for the wireless network using the flash drive. In one implementation, each device configuration log file is identified by a file name that contains the last 8 bytes of the MAC address of the provisioned device in ASCII-HEX format. This file name allows the computer on which the network settings are created to identify the provisioned device.

[0047] In an embodiment of the invention, the USB flash drive 508 further stores an autorun file, such as wireless.cfg 518. When the USB flash drive 508 is attached to a compatible device that recognizes the autorun file, the detection of wireless.cfg 518 automatically triggers the device to execute the network setup program 510. In this way, no user intervention is required to transfer the network settings to the device after the USB flash drive 508 has been attached.

[0048] A method of using a portable media device to configure a thin client device in accordance with an embodiment of the invention is now described with reference to FIG. 6. This method is used with, for example, thin client or "headless" devices such as wireless printers, digital audio receivers and smart displays. The described method simplifies the configuration process for such devices in comparison to previous methods. The method begins at step 602 with the attaching of a portable media (PM) device, such as a USB flash drive, to a personal computer that is used to create the settings for the thin client device. The user then

runs a PC-based configuration tool on the personal computer at step 604. Through an interactive process with the aid of user interface screens, the configuration tool helps the user create the configuration settings for the intended thin client device at step 606. The configuration settings are then written into the portable media device as one or more XML files at step 608. The portable media device is then removed from the personal computer and installed into the new computing device to be configured at step 612. The computing device uploads the relevant network configuration settings from the PM and is provisioned at step 614. If the computing device supports an automatic discovery protocol, such as UPnP, a computer on the network will immediately recognize the new device and be notified of its availability. In one embodiment, the uploading is performed automatically, without any user intervention. The computing device searches the XML files stored on the portable media device for particular data fields it requires, and uploads the corresponding data from the portable media device. In this way, the device only reads data relevant to its operation. The device then executes a configuration program to automatically configure itself using the received configuration setting data.

[0049] Once the configuration settings are implemented on the device, the device preferably signals to the user that it has been successfully configured. How the device signals the completion of configuration would depend on the signaling means the device has. Most thin client devices have one or more light-emitting diodes (LED's), which may be advantageously used for this purpose. For instance, the thin client device may flash an LED three times with a 1-second on/off cycle. Alternatively, if the thin client device has an LCD screen, a simple message may be displayed thereon to indicate to the user that configuration is completed. As another example, a wireless picture frame 268 (FIG. 2) is a device that receives data for a digital picture wirelessly from a source (e.g., a PC) and has a display area for displaying the digital picture. When it is configured using the portable media device, it can use the display area to display the configuration completion message.

[0050] In one embodiment, the thin client device that has been configured writes a configuration log file back into the portable media device at step 618. This configuration log file allows the user to ascertain that the device has the proper configuration settings and can be used for diagnostic purposes. The method continues at step 620 with the user determining

whether an additional computing device is to be configured using the data on the portable media device. If so, the portable media device is attached to the new computing device at step 612. Any number of new computing devices can be added to the network in this manner. If there is no further computing device to be configured, the user returns to the personal computer and installs the portable media device in the computer. The computer then displays (see FIG. 4F) the configuration data from the configuration log files of the computing devices that have been configured using the portable media device.

[0051] In one embodiment, the portable media device includes a time-to-live (TTL) field, and devices configured using the portable media device are denied access to the network after the prescribed TTL has passed. In this way a device is granted only temporary access to the network. In another embodiment, the TTL field is used in conjunction with an authentication scheme, such as a thumbprint. In this way, authenticated devices are granted permanent access to the network while guests are granted only temporary access as determined by the TTL field.

[0052] FIG. 7 presents an exemplary sequence of actions illustrating the generating and transferring of network configuration settings, by way of a portable media device, in order to create an ad hoc wireless network. As illustrated in FIG. 7, a user 700 interacts with an initiating computer 710 to create the network settings. The network settings preferably include a Name string used to identify the network, such as an SSID string used during the 802.11 beaconing process, and a Pre-Shared Key string used for network authentication. Authentication types could include, for example, WEP, WPA PSK, or 802.11i PSK encryption. The key format could be a binary or hexadecimal number, an alphanumeric string, or a keyword. The settings are stored on a portable media device 720 as files in the XML format. The portable media device 720 is then used to configure another peer device 730 of the wireless ad hoc network.

[0053] In one embodiment, previously uploaded configuration settings are stored as multiple profiles. By using profiles, users can easily revert to previous network settings and switch to other networks without needing to obtain the network configuration settings again.

[0054] FIG. 8 presents an exemplary sequence of actions illustrating the generating and transferring of network configuration settings, by way of a portable media device, in order to configure a wireless access point (WAP) used to provide 802.11 wireless network connectivity.

In FIG. 8, the user 800 runs a network setup wizard on a personal computer 810 to enter the necessary settings to generate the XML file stored on the portable media device 820. The personal computer 810 generates the XML file and instructs the user regarding the use of the portable media device to transfer network settings to a new device, in this case the wireless access point 830. As illustrated, the portable media device 820 is a USB flash drive (UFD), but it could be representative of any media or transfer mechanism for moving XML-format network configuration settings between a PC and a device such as the wireless access point.

[0055] In the process of creating network settings for configuring computing devices, the personal computer generates the WSETTINGS.XML, LSETTINGS.XML, and NSETTINGS.XML files according to corresponding XML schemas. FIG. 9 illustrates a conceptualization of a schema 900 corresponding to the XML file used in an embodiment for storing the ad hoc wireless network settings on the portable media device. Each element of schema 900 represents a wireless configuration setting. Service set identifier (SSID) 910 is a 1-32 byte string representing the name of the wireless network. SSID 910 can only occur once. Network Key 915 is a string that the PC will automatically generate, or alternatively, receive from the PC user. The Network Key 915 is used for encryption on the wireless network. Authentication Type 920 indicates the authentication protocol used by the wireless network. Authentication Type 920 allows a range of possible values, including open, shared, WiFi Protected Access (WPA), WPA Pre-Shared Key (PSK), WPA-none, WPA2, or WPA2 PSK. Encryption Type 925 indicates the encryption protocol used by the wireless network. Encryption Type 925 allows a range of possible values, including none, Wireless Encryption Protocol (WEP), Temporal Key Integrity Protocol (TKIP), and Advanced Encryption Standard (AES).

[0056] Type 930 indicates a connection type, and may have as its value either extended service set (ESS) in the case of an ad hoc network, or infrastructure basic service set (IBSS) in the case of an infrastructure network. Key Index 935 indicates the location of the specific key used to encrypt messages, and may have a value of 1, 2, 3, or 4. Key Index 935 is used with WEP. Key Provided Info 940 indicates whether a key is provided automatically, and can have a value of either 0 or 1. The "802.1X" field 945 indicates whether IEEE 802.1X protocol is used on the network, and can have a value of either 0 or 1. 2.4 GHz Channel 950 indicates

which 2.4 GHz Channel, if any, is being used by the wireless network, and can have a value in the range of 1 to 14. 5 GHz Channel 955 indicates which 5 GHz channel, if any, is being used by the wireless network, and can have a value of 36, 40, 44, 48, 52, 56, 60, 64, 149, 153, 157, or 161. WAP Mode 960 indicates the mode in which the wireless access point is operating.

WAP Mode 960 can have a value of infrastructure, bridge, repeater, or station.

[0057] An exemplary XML schema for wireless network settings is as follows:

```
<?xml version="1.0" encoding="utf-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.microsoft.com/provisioning/WirelessProfile/2004"
xmlns="http://www.microsoft.com/provisioning/WirelessProfile/2004"
elementFormDefault="qualified"
version="1">
<xs:element name="wirelessProfile">
<xs:complexType>
<xs:sequence>
<xs:element name="config">
<xs:complexType>
<xs:sequence>
<xs:element name="configId" minOccurs="1" maxOccurs="1">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:length value="36" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="configHash" minOccurs="0" maxOccurs="1">
<xs:simpleType>
<xs:restriction base="xs:hexBinary">
<xs:length value="40" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="configAuthorId" minOccurs="1" maxOccurs="1">
<xs:simpleType>
```

```

        <xs:restriction base="xs:string">
            <xs:length value="36" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="configAuthor" minOccurs="1" maxOccurs="1">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="128" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="ssid" minOccurs="1" maxOccurs="1">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="32" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="connectionType" minOccurs="1" maxOccurs="1">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="IBSS" />
            <xs:enumeration value="ESS" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>

<xs:element name="channel2Dot4" type="xs:integer" minOccurs="0" maxOccurs="1" />
<xs:element name="channel5Dot0" type="xs:integer" minOccurs="0" maxOccurs="1" />
<xs:element name="deviceMode" minOccurs="0" maxOccurs="1">
    <xs:simpleType>
        <xs:restriction base="xs:string">

```

```

        <xs:enumeration value="infrastructure" />
        <xs:enumeration value="bridge" />
        <xs:enumeration value="repeater" />
        <xs:enumeration value="station" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="primaryProfile" type="profileInstance" minOccurs="1" maxOccurs="1" />
<xs:element name="optionalProfile" type="profileInstance" minOccurs="0"
    maxOccurs="unbounded" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:complexType name="profileInstance">
    <xs:sequence>
        <xs:element name="authentication" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="open" />
                    <xs:enumeration value="shared" />
                    <xs:enumeration value="WPA-NONE" />

                    <xs:enumeration value="WPA" />
                    <xs:enumeration value="WPAPSK" />
                    <xs:enumeration value="WPA2" />
                    <xs:enumeration value="WPA2PSK" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element name="encryption" minOccurs="1" maxOccurs="1">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:enumeration value="none" />
                    <xs:enumeration value="WEP" />
                    <xs:enumeration value="TKIP" />
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

```

```

        <xs:enumeration value="AES" />
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="networkKey" minOccurs="1" maxOccurs="1">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="64" />
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="keyIndex" type="xs:integer" minOccurs="0" maxOccurs="1" />
<xs:element name="keyProvidedAutomatically" type="xs:boolean" minOccurs="1" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:schema>

```

[0058] An exemplary instance of a WSETTINGS.XML file is set forth below. This instance was generated, for example, by the PC using the wireless configuration XML schema set forth immediately above.

```

<?xml version="1.0" ?>
<WirelessProfile
  xmlns="http://www.microsoft.com/provisioning/WirelessProfile"
  >
    <ssid>HOMENET</ssid>
    <ConnectionType>ESS</ConnectionType>
    <Authentication>WPAPSK</Authentication>
    <Encryption>TKIP</Encryption>
    <NetworkKey>WirelessKey!0</NetworkKey>
    <KeyProvidedAutomatically>0</KeyProvidedAutomatically >
    <IEEE802.1Xenabled>0</IEEE802.1Xenabled>
</WirelessProfile>

```

[0059] FIGS. 10A-C illustrate a conceptualization of the schemas 1010, 1020, and 1030 corresponding to the NSETTINGS.XML file for WAN configurations. Schema 1010 defines the NSETTINGS.XML file for a Dynamic Host Configuration Protocol (DHCP) WAN. Schema 1020 defines the NSETTINGS.XML file for a WAN using static IP addressing. Schema 1030 defines the NSETTINGS.XML file for a Point-to-Point Protocol over Ethernet (PPPoE) WAN.

[0060] Schema 1010 includes the following elements: Hostname 1011, DNS IP Auto Pushed 1012, Clone MAC Address 1013, MAC Address 1014, DNS1 IP Address 1015, and DNS2 IP Address 1016. Hostname 1011 defines a string signifying the name of the DHCP host. DNS IP Auto Pushed 1012 indicates whether a domain name service IP address is automatically pushed to the DHCP client. Clone MAC Address 1013 indicates whether the Media Access Control (MAC) address of the DHCP client should be cloned for use in the WAN, and takes a value of 0 or 1. MAC Address 1014 defines the MAC address of the DHCP client. DNS1 IP Address 1015 and DNS2 IP Address 1016 define IP addresses for domain name servers.

[0061] Schema 1020 includes the following elements: IP Address 1021, IP Subnet 1022, IP Subnet Mask 1023, DG 1024, DNS1 IP Address 1025, and DNS2 IP Address 1026. IP Address 1021 defines the static IP address of the device to be configured. IP Subnet 1022 defines the IP subnet of the device to be configured. IP Subnet Mask 1023 defines the IP subnet mask of the device to be configured. DG 1024 defines the default gateway for the device to be configured. DNS1 IP Address 1025 and DNS2 IP Address 1026 define IP addresses for domain name servers.

[0062] Schema 1030 includes the following elements: Username 1031, Password 1032, Service Name 1033, Max Idle Time 1034, and Auto-Reconnect 1035. Username 1031 defines a username for a device to be configured, and Password 1032 defines a password for a device to be configured. Service Name 1032 defines a string indicating the name of a PPPoE server. Max Idle Time 1034 defines a maximum idle time for the device to be configured. Auto-Reconnect 1035 indicates whether the device to be configured should automatically reconnect to the network, and takes a value of 0 or 1.

[0063] FIG. 11 illustrates a conceptualization of the schema 1100 corresponding to the LSETTINGS.XML file for LAN configuration. Schema 1100 includes the following elements: Workgroup Name 1110, IP Settings 1120, IP Subnet 1130, and Network Shares 1140. Workgroup Name 1110 defines a workgroup name for the LAN. IP Settings 1120 defines the IP settings for the LAN device. The IP Settings 1120 may include subelements DHCP 1121 for defining DHCP settings, Static IP 1122 for defining settings using static IP addressing and IP Subnet 1123 for defining IP subnet settings associated with the Static IP 1122 sub-element. IP Subnet 1130 defines an IP subnet for the device to be configured, wherein the IP subnet defined by IP Subnet 1130 may be different from the IP subnet defines in the IP Subnet 1123 sub-element of the IP Settings 1120, for example, IP Subnet 1130 may define an alternative and/or complementary IP subnet such as a public IP subnet if the IP Subnet 1123 defines a private IP subnet or vice versa. Network Shares 1140 defines the file sharing configuration for the LAN which the device to be configured is to join, including network drive mappings.

[0064] It will be appreciated that an improved system and method for configuring thin client computing devices have been disclosed herein. In view of the many possible embodiments to which the principles of the present invention may be applied, it should be recognized that the embodiments described herein with respect to the drawing figures are meant to be illustrative only and should not be taken as limiting the scope of the invention. For example, those of skill in the art will recognize that the illustrated embodiments can be modified in arrangement and detail without departing from the spirit of the invention. Although the invention is described in terms of software modules or components, those skilled in the art will recognize that such may be equivalently replaced by hardware components. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.